Bolt Beranek and Newman Inc.

LEVEL II (12)

Report No. 4564

Development of a Voice Funnel System

Quarterly Technical Report No. 7
1 February 1980 to 30 April 1980

January 1981

Prepared for:
Defense Advanced Research Projects Agency

UNCLASSIFIED (9) Quarterly technical rept. no. 7 1 Feb - 30 Apr 81

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: 4564	2. GOVT ACCESSION NO.: AD-A094 036
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): (4) DEVELOPMENT OF A VOICE FUNNEL SYSTEM. QUARTERLY TECHNICAL REPORT NO. 7	5. TYPE OF REPORT & PERIOD COVERED: Quarterly Technical
	6. PERFORMING ORG. REPORT NUMBER: 4564
7. AUTHOR(s): (10) R. D. Rettberg (15)	8. CONTRACT OR GRANT NUMBER(s): MDA903-78-C-0356, ARPA Order-3653
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc. 50 Moulton Street, Cambridge, MA 02238	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order No. 3653
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency 1400 Wilson Blvd., Arlington, VA 22209	12. REPORT DATE: (11) January 1981
	13. NUMBER OF PAGES: 26 (12) 30
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (14) BBN-4564	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Voice Funnel, Digitized Speech, Packet Switching, Butterfly Switch, Multiprocessor

This report

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This Quarterly Technical Report covers work performed during the period noted on the development of a high-speed interface, called a Voice Funnel, between digitized speech streams and a packet-switching communications network.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060100

DEVELOPMENT OF A VOICE FUNNEL SYSTEM

QUARTERLY TECHNICAL REPORT NO. 7
1 February 1980 to 30 April 1980

January 1981

This research was sponsored by the
Defense Advanced Research Projects
Agency under ARPA Order No.: 3653
Contract No.: MDA903-78-C-0356
Monitored by DARPA/IPTO
Effective date of contract: 1 September 1978
Contract expiration date: 31 January 1981
Principal investigator: R. D. Rettberg

Prepared for:

Dr. Robert E. Kahn, Director
Defense Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Boulevard
Arlington, VA 22209

Table of Contents

FIGURES

1. Introduction

This Quarterly Technical Report, Number 7, describes aspects of our work performed under Contract No. MDA903-78-C-0356 during the period from 1 February 1980 to 30 April 1980. This is the seventh in a series of Quarterly Technical Reports on the design of a packet speech concentrator, the Voice Funnel.

This report describes the hardware design of the Input/Output system to support the Voice Funnel.

2. I/O Module Hardware Description

The design of the I/O system for the Voice Funnel offers many design choices. A line must be walked between accepted structures and novel approaches, between high performance and high cost, and between complex configurations and oversimplification. This report deals with how that line has been followed in the design of the I/O Module, which provides high bandwidth I/O for the Voice Funnel.

Two dominant issues motivated the architecture of the I/O Module: flexibility and performance. Flexibility is important because of the environment into which the I/O Module must fit. The Voice Funnel is part of the Wideband Satellite Experiment, which involves research in data communications protocols. The hardware of the Voice Funnel must be flexible enough to accommodate this research. Performance is important so that the multiprocessor approach to packet switching can be tested unencumbered by low throughput limitations in the I/O system. As the environment of the Voice Funnel stabilizes, it is possible that a simpler, less flexible I/O system may be developed.

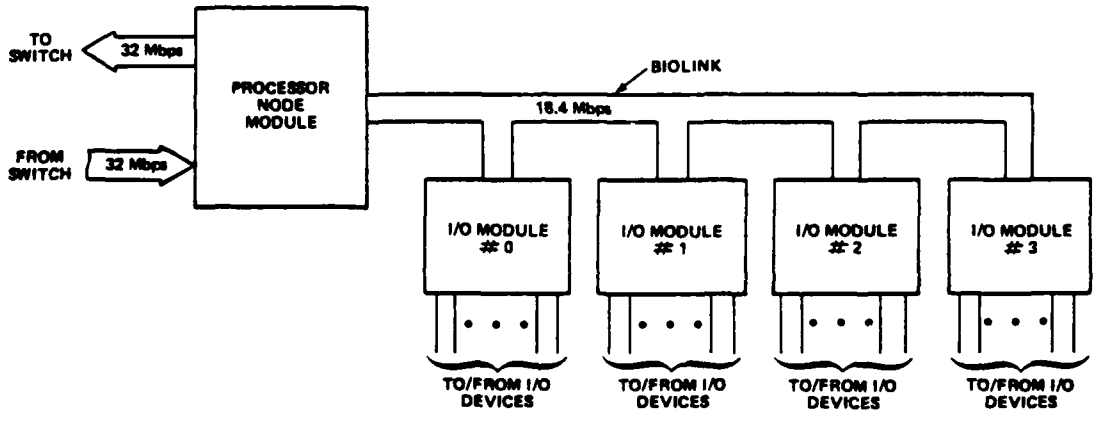
These goals are achieved by incorporating a custom 16-bit microprogrammed processor on each I/O Module. This processor will perform such functions as direct memory access, input and output queueing, time stamping of packets, and control block chaining. High throughput is realized by having the MC68000 set

up chains of control blocks which are then used by the I/O Modules to control the transfer of data to and from Processor Node memory. In some cases, the I/O Module processor may also participate in the line protocol; one such example is the interface to the Pluribus Satellite IMP (PSAT).

An I/O module occupies one 12 x 17" printed circuit board. It supports four asynchronous and four synchronous channels with an aggregate data rate of four megabits per second (4 Mbps). Up to four I/O modules can be attached to one Processor Node as shown in Figure 1 for a total of sixteen asynchronous and sixteen synchronous channels with a total aggregate data rate of 16 megabits per second. These modules communicate with the Processor Node over an I/O Bus called the BIOLINK.

2.1 I/O Controller Candidates

Two architectures were investigated for the controller in the I/O module: a commercial I/O microprocessor and a custom microprogrammed processor. The advantages of a commercial microprocessor are its small size, low cost, and multiple sourcing. Advantages of a custom microprogrammed processor are that it can provide additional I/O processing which otherwise would have to be performed by the central processor and can potentially provide higher throughput.



Maximum I/O System on a Processor Node
Figure 1

The Intel 8089 I/O processor was examined for use as the I/O Module controller. This processor performs the function of an intelligent Direct Memory Access (DMA) controller. Since it can operate completely in parallel with the MC68000, it should give good throughput in a highly I/O intensive application such as the Voice Funnel.

Unfortunately, an examination of this I/O controller immediately uncovered two major problems. The first was that the 8089's two DMA channels support only one bidirectional communications link. Thus an I/O Module would require many of these 8089 controllers. This would be acceptable if the 8089 did not require any support circuitry. However, since a minimum 8089 controlled communications link requires approximately 8-10 integrated circuits, the original simplicity advantage is lost.

The second problem is how to obtain efficient information transfers between the I/O Controller and the Processor Node's memory. The 8089 is byte-oriented, and therefore cannot take advantage of the BIOLINK 16-bit data path to the Processor Node's memory, losing half the potential bandwidth. In addition, the I/O controller faces memory contention on the Processor Node. It should have either flow control or a large buffer to accommodate transfer delays to memory.

The BIOLINK bandwidth and latency constraints can be solved by transferring two bytes of data in each BIOLINK DMA transfer

and providing the equivalent of a data FIFO on each synchronous protocol chip. For example, transmission of a message might not start until at least 32 bytes of data reside in the 8089's local memory. When this condition is satisfied, the 8089 would startup the line protocol chip loading a byte of data from the local buffer into the chip's input buffer each time it became empty. Simultaneously, the DMA facility would transfer data from the Processor Node's local memory to the 8089's local memory. Unfortunately, the 8089 has insufficient processing power to perform the buffering.

Performance is especially important in the two links to the PSAT. Since the 8089 approach would not support the bandwidth of these links, a separate interface would be required. Due to these problems, we decided to develop a custom microprocessor, which will support the high speeds and flexibility requirements of the Voice Funnel environment.

2.2 Microinterrupts

The I/O controller operates at a microinstruction execution rate of 8 MHz. Since the maximum rate of the asynchronous channels is 38.4 kbps, these channels can be completely serviced by means of polling in the I/O controller. The synchronous channels operate at rates up to 2 MHz, too fast to operate on a strictly polled basis. Polling is still used to set up a

transfer between the synchronous protocol chip and scratchpad memory and/or between scratchpad memory and Processor Node local memory, but data transfers then occur via a microinterrupt system. When a data transfer is to take place, the background polling is interrupted, and a service routine is executed which performs the data transfer. All the polling routines are designed to allow a microinterrupt to occur within a few microinstructions. Thus a very low delay between event occurrence and event service is achieved (maximum of 0.375 microseconds).

Five events may trigger a microinterrupt:

1. The MC68000 reads or writes of one of the control/data registers associated with each of the 8 channels.
2. One of the four synchronous protocol chips has a new byte of data to insert in its input FIFO.
3. One of the four synchronous protocol chips is ready to be loaded with a new data or control word from an output FIFO.
4. One of the four output FIFOs needs to be loaded with a data word from the Processor Node's local memory.
5. Data from one of the input FIFOs needs to be loaded into the Processor Node's local memory.

Each of these microinterrupt events can be enabled or inhibited by the I/O controller. For example, after the last word has been read from the Processor Node's local memory and loaded into an output FIFO, the microinterrupt service routine sets a bit to inhibit further microinterrupts for that channel.

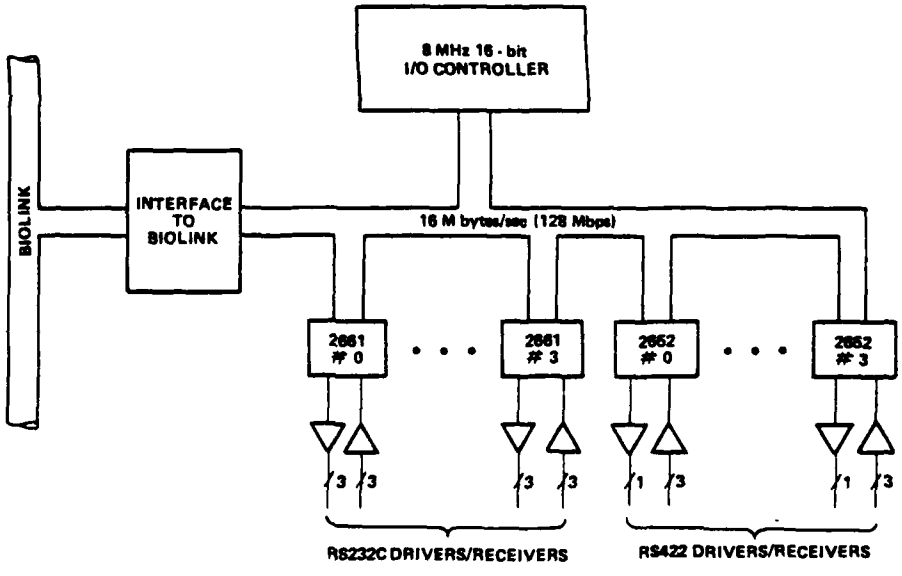
Sometime later, the polling routine notices that more blocks remain in the chain. It then sets up the next block's DMA parameters and clears the bit to enable microinterrupts for the channel. The polling approach introduces a delay between the completion of one DMA transfer and the initiation of the next. However, this delay is small (approximately 2 microseconds), and because of the FIFOs provided, does not introduce any throughput loss for blocks of moderate size (i.e. more than 6 bytes).

2.3 I/O Controller Hardware Implementation

A block diagram of the I/O Module is shown in figure 2. It includes four Signetics 2661 Enhanced Programmable Communications Interface (EPCI) chips with RS232C line driver/receivers, four Signetics 2652 Multi-Protocol Communications Controllers (MPCC) chips, with RS422 line driver/receivers, an interface to the BIOLINK, and a 16-bit 8MHz microinterrupt-driven microprocessor with 512 64-bit words of control store and 2K bytes of scratchpad memory.

2.3.1 I/O Controller

Within the I/O Controller there are four parts interconnected on an internal 16-bit bus. This bus is isolated from the bus attached to the communications chips to allow I/O Controller processing while accessing the slower communications



I/O Module Architecture
Figure 2

chips. Gating between the two buses allows a variety of needed byte operations to be performed.

A 16-bit bipolar microprocessor unit using the Advanced Micro Device 2901 provides the means for data manipulations and storage for 17 16-bit variables used in the polling and microinterrupt service routines. Distinctive characteristics of the microprocessor are:

- A two-address architecture allows simultaneous access to two working registers.
- An eight function ALU performs addition, subtraction in either operand order, and five logic functions between two sources.
- Flexible data source selection allows ALU data to be selected from five source ports for a total of 203 source operand pairs for every ALU function.
- Left/right 32-bit rotates are provided.
- Carry, zero, negative, and shift output status flags are provided for both polling and microinterrupt environments.

A 512 word 64-bit read-only Control Store memory supplies the microinstructions which both drive the microprocessor and control all other I/O Module resources. To enhance throughput, the next microinstruction is fetched from Control Store while the current microinstruction is being executed. This standard pipelining approach would permit the I/O Controller to operate at a 9MHz microinstruction rate. A writeable Control Store would have been preferred if not for its cost (a factor of nine over non-writeable Control Store) and the additional complexity

incurred in loading the Control Store over communications links controlled by an I/O Controller.

The 9-bit Control Store address is sourced by a conventional microprogram sequencer in both polling and microinterrupt routines. The sequencer is implemented using the Advanced Micro Devices 2911 augmented by two multiplexors which allow two-condition four-way branching. Other microsequencer features include a pushdown stack for saving up to 4 return addresses, a means for returning to the zero microcode word on system reset, and an internal address register which can be used for the address of a commonly used routine.

The microinterrupt structure in the I/O Module is relatively complicated in order to provide very low event response latency. The various elements in the microinterrupt structure both arbitrate which channel is to be serviced and generate the service routine address of the highest priority event. Round-robin arbitration is used to select which input and output FIFOs are to be serviced. In this way, all channels are treated identically in their access to the Processor Node's local memory.

A 1K X 16 scratchpad memory provides control variable and data storage for all the channels. This memory is partitioned into five blocks:

1. 32 control registers for each asynchronous protocol chip

2. 64 bytes of input FIFO for each asynchronous protocol chip
3. 64 control registers for each synchronous protocol chip
4. 64 words of input FIFO for each synchronous protocol chip
5. 64 words of output FIFO for each synchronous protocol chip

A very simple form of scratchpad management hardware is included. This hardware allows the microcode to change from a group of registers associated with one communications chip to a group associated with another communications chip during a microinstruction. In addition, a pushdown stack is included to save the working environment when a microinterrupt occurs. Indirect addressing is provided to allow access to FIFO ring buffers entries.

2.3.2 BIOLINK Interface

With the I/O Controller busy servicing the communications chips, it is important to reduce the overhead associated with conforming to the BIOLINK bus protocol. Although this protocol is simple and synchronous with the I/O Controller, directly controlling this protocol could still consume a substantial portion of the I/O Controller's bandwidth. We have provided a simple finite state machine (FSM) to assist the I/O Controller in its interaction with the BIOLINK. In this architecture, the I/O

Controller sets up the data operation and activates the BIOLINK controller FSM. On completion, the FSM signals the I/O Controller of the operation's completion.

2.4 Asynchronous Communication Channels

For the four asynchronous channels, the I/O Module features:

- Speed: Each of the four channels may run at program selectable speeds up to 38400 bps.
- Flexibility: Complete program control of each line for:
  - Data rate -- 16 standard speeds
  - Character size -- 5,6,7,or 8 bits
  - Stop code length -- 1, 1.5, or 2 bits
  - Odd, even, or no parity generation and checking
  - Interrupt vectors on up to three separate events;
- Program controlled hardware echo of received characters;
- Local or remote maintenance loop back mode;
- Full or half duplex operation;
- DMA transmitter for each channel, with byte count and address registers;
- Break detection and program controlled break generation;
- Complete set of line conditioning signals available for interfacing to various modems;
- 16-character FIFO for received characters;
- RS232C compatible inputs and outputs;
- Programmable order of bytes within a word.

2.4.1 Asynchronous Receiver Operation

Reception on each of the four asynchronous input channels is effected by means of a Signetics 2661 Enhanced Programmable Communications Interface (EPCI). These MOS/LSI circuits perform all the necessary functions for double-buffered asynchronous character assembly.

At least once each 250 microseconds, the I/O Controller examines the status of each EPCI. If it notices that a character has been assembled, it loads both the status and data byte into either a holding register or 16 entry FIFO. It also adjusts internal status bits for monitoring by the MC68000.

The I/O Controller issues an interrupt request to the MC68000 on the occurrence of various events. Serializing the possible interrupt requests from one I/O Module is accomplished by allowing only one interrupt to be pending at a time, and by scanning from the last interrupting channel.

Character length, parity sense, number of stop bits, etc., used by the EPCI to perform character assembly and error reporting are stored both within the EPCI and the I/O Controller. The shadow copies in the I/O Controller allow the MC68000 to both write and read these control registers, which would otherwise be write only.

2.4.2 Asynchronous Transmitter Operation

Transmission on each asynchronous channel is also effected by means of the EPCIs. These devices perform all the necessary functions for double-buffered character transmission.

There are two mechanisms by which the MC68000 can cause a sequence of characters to be transmitted. In the first technique, the MC68000 loads each character into a holding register in the EPCI after determining that the holding register is empty. The holding register status can easily be determined either by interrogating the EPCI status or as a result of an interrupt caused by the holding register becoming empty.

In the second technique, the MC68000 sets up a block in the Processor Node's memory containing the sequence of characters and loads the starting address and character count into I/O Controller registers. The I/O Controller then assumes responsibility for transmitting all characters in the block. After the last character is loaded into the EPCI, the I/O Controller updates an internal status bit which can cause an MC68000 interrupt.

Data bits are transmitted least significant bit first. When the I/O Controller transmits the characters from a block in Processor Node memory a decision is made as to which byte within a 16-bit word to transmit first. To provide the maximum flexibility, the programmer may specify the order of the bytes.

As was done with the receiver channels, the number of stop bits, number of bits in a character, parity inclusion and sense etc., are loaded by the MC68000 into both the EPCI and I/O Controller internal registers so that they can be both read and written by the PNC.

2.4.3 Modem Interface Signals

In addition to a receiver and a transmitter section, each EPCI provides two inputs and two outputs for interfacing to modems.

The modem interface signals provided are: Data Carrier Detect, Data Set Ready, Request to Send, and Data Terminal Ready. If enabled, any change on the Data Carrier Detect or Data Set Ready lines sourced by the modem will cause a interrupt request to the MC68000.

All leads (including receiver input and transmitter output) conform to the EIA RS232C and CCITT electrical specifications.

2.5 Synchronous Communication Channels

For the four synchronous communications channels, the I/O Module features:

- Speed -- data rates up to 2 megabits per second;
- Flexibility -- complete program control of each line for:
  - Sync or secondary station address comparison;
  - Idle transmission of SYNC/FLAG or MARK;
- Automatic detection and generation of special bit-oriented protocol FLAG, ABORT, and GA control sequences;
- RS422 compatible inputs and outputs;
- Maintenance mode for self testing;
- 64 character input and output FIFOs to improve latency;
- DMA controller for both input and output with aggregation of characters into words to reduce BIOLINK bandwidth usage;
- Chained DMA control blocks to provide fast buffer swaps without MC68000 intervention;
- Programmable interval timer to minimize the time between transmitted messages;
- Time stamping of incoming messages;
- Transmission of messages at a given time of day.

2.5.1 Bit-Oriented Protocol Messages

The I/O Module supports bit-oriented messages which are transmitted in frames. A frame consists of an 8-bit FLAG sequence, followed by an 8-bit ADDRESS sequence, an INFORMATION sequence, a 16-bit FRAME CHECK sequence, and finally another FLAG sequence. This industry standard Bit Oriented Protocol (BOP) features:

- Independence of codes, line configuration, and peripherals;
- Positional significance instead of control characters or character counts;
- Information transparency through zero insertion and deletion;
- Error checking on a complete frame.

The INFORMATION field may vary in length including a different length in sequential frames making up a complete transmission. While the BOP allows this field to be any number of bits, the I/O Controller hardware limits the INFORMATION field to be only non-zero multiples of 8-bits.

Transparency is effected by inserting a zero bit following any sequence of five one bits in the data. This means that no data in the message can look like the FLAG sequence, which consists of a sequence of six one bits. This necessarily implies a communication link bandwidth less than the rated bit rate by up to 17%.

A FRAME CHECK sequence (FCS) is included in all BOP frames to detect errors which may occur during transmission. Content of this field is based on a cyclic redundancy check using the CRC-CCITT generator polynomial:

$$G(X) = X^{16} + X^{12} + X^{5} + 1$$

Although not part of the BOP, the time interval between successive frames becomes important when interfacing to external

systems. The BOP allows the ADDRESS sequence of one frame to occur as soon as 8 bit intervals after the last bit in the FRAME CHECK sequence of the previous frame. At the highest data rates, we would expect an external system to load the INFORMATION field into its local memory via DMA. For a two megabit per second link, only 4 microseconds is allowed to swap between data buffers associated with successive frames. Except where dedicated buffer swapping hardware or relatively deep input FIFO buffers are employed by the external system, this minimum interval between two successive frames must be extended to accommodate the processing speed of the receiver station processor.

A time-of-day clock (incremented each 62.5 microseconds) is used to generate a minimum time between successive frames. Sampling this clock at random intervals can introduce a one clock tick uncertainty. Thus to guarantee at least 62.5 microseconds between frames, a two clock interval must be specified. The coarseness of the clock can substantially reduce the communication link bandwidth utilization at high data rates, short frames, and minimum interval between frames. It is assumed however that if such high bandwidth is required, the external system is very fast and will have the same type of FIFO buffering provided by the I/O Controller, and will therefore not require any extra time between frames.

In the Voice Funnel application, the times at which a frame is received and transmitted are important. In a process oriented

system such as is envisioned for the Voice Funnel environment, real time jobs require substantial system overhead to acquire a time sense. The I/O Controller helps by providing a time-stamp on incoming frames and allowing transmission of frames at a specified time of day. While the use of this approach to packet switching is somewhat experimental, we expect a substantial reduction in the MC68000 per-packet processing time.

2.5.2 Synchronous Receiver Operation

Frame reception on each of the four synchronous input channels is effected by means of a Signetics 2652 Multi-Protocol Communications Circuit (MPCC). These MOS/LSI chips perform all the necessary functions for double-buffered BOP frame receipt.

Associated with the receiver section of each MPCC is a 64-word memory block, miscellaneous address, count, and control registers, and three hardware flags which interact with the I/O Controller's microinterrupt structure.

The 64-word block is used as a ring buffer implementing an input FIFO. One of the hardware flags interacts with the microinterrupt structure to inhibit data exchanges from MPCC to input FIFO when the input FIFO becomes full. Since in normal operation we never expect the input FIFO to fill, this flag provides a mechanism for using the MPCC's overrun error detection

logic to indicate that the I/O Controller has not kept up with the channel.

Another hardware flag reflects the number of data bytes in the input FIFO. It indicates that the input FIFO is not empty if 1) the input FIFO contains at least two bytes of data, or 2) the input FIFO contains only one entry but that entry resulted from receipt of the last byte in the frame. This flag, together with the flag indicating that the DMA channel is currently set up, cause a microinterrupt service routine to be executed which aggregates two bytes of data into a word which is then loaded into the Processor Node's local memory.

What becomes clear from this entire discussion is that special hardware is required to provide both an input FIFO and DMA channel. It is surprising that the protocol chip designers have not recognized the need for deep FIFOs on both input and output.

2.5.3 Synchronous Transmitter Operation

Transmission on each synchronous line is also effected by means of MPCCs. These chips perform all the necessary functions for double-buffered BOP frame serialization.

Associated with the transmitter section of each MPCC are a 64-word scratchpad block, miscellaneous address, count, and

control registers, and three hardware flags which interact with the I/O Controller's microinterrupt structure.

The 64-word block is used as a ring buffer implementing an output FIFO. Three hardware flags interact with the microinterrupt structure in much the same way as the three flags in the synchronous receiver side to control loading Processor Node memory data into the output FIFO and FIFO data into the MPCC.

At the end of each message, the I/O Controller normally inhibits the transmitter section, even though one or more characters from the next frame to be transmitted are already in the output FIFO. In polling each channel for something to set up, the I/O Controller will notice that both the inter-frame time interval and time-of-day transmission time have been satisfied, and that at least a minimum number of entrys reside in the output FIFO. It then starts up the MPCC for transmission and enables the microinterrupt structure to take over unloading the output FIFO into the MPCC.

2.5.4 Control Block Architecture

From a software point of view, control of synchronous data transfers occurs as a result of appending and inserting control blocks to a control block queue. Each control block refers to a

buffer which has three parts: the Supervisor Parameter Block (SPB), the User Parameter Block (UPB), and the Data Block (DB). A queue is formed by linking SPBs together with forward and backward pointers. The SPB contains those parameters which, if inadvertently modified by the user program, could cause severe system problems. Examples of such privileged variables are the physical addresses of the UPB and DB, the buffer's maximum length, and interrupt vectors to be used when error events are detected. The UPB contains those parameters needed by the I/O Controller which will not cause a system failure if modified or incorrectly set up by a user program. The DB is that memory area in which received frame data is stored or from which transmitted data is fetched.

The queue of buffers provides two important advantages over conventional intervene-at-end-of-frame techniques. First, it reduces the minimum inter-frame period to that required by the I/O Controller rather than that required by the MC68000's worst case response time to an end-of-frame event. Not only is the I/O Controller an order of magnitude faster than the MC68000, but there may be up to four I/O Controllers for each MC68000. By having the MC68000 set up the frames well in advance of their transmission, the real time processing requirements on the MC68000 will be limited by average processing per frame rather than peak frame processing rates.

Second, the queue should reduce the system overhead per frame. In conventional systems, it is difficult to avoid using a majority of the processing time per frame in context switches, identity determination, and privilege checking. By fully using the queue mechanism, many frames may be processed in one MC68000 intervention, thereby reducing the average system overhead per frame.

The incorporation of a high performance, programmable I/O system is making the design of the Voice Funnel much simpler. Since the I/O system is performing the frequent, low latency functions, the main processors of the Voice Funnel can be programmed more straightforwardly, and can largely ignore the difficulties of high bandwidth I/O. In addition, the provision of a fast, sophisticated I/O system will permit the Voice Funnel to operate at much higher throughput rates than would otherwise be possible.

DISTRIBUTION OF THIS REPORT

Defense Advanced Research Projects Agency

Dr. Robert E. Kahn (2)

Defense Supply Service -- Washington

Jane D. Hensley (1)

Defense Documentation Center (12)

Bolt Beranek and Newman Inc.

Library

Library, Canoga Park Office

R. Bressler

R. Brooks

P. Carvey

P. Castleman

G. Falk

F. Heart

M. Hoffman

M. Kraley

W. Mann

J. Pershing

R. Rettberg

E. Starr

E. Wolf